

A NEW SPECTRAL PARAMETER GRADIENT VECTOR THAT SATISFIED WEAK SECANT EQUATION FOR SOLVING LARGE SCALE UNCONSTRAINED OPTIMIZATION

M.I. ISHAK¹, S.M. MARJUGI, AND L.W. JUNE

ABSTRACT. This paper presents a new method called LBFGS-QM for line search selection in the frame of spectral L-BFGS method. Choosing a good line search can lead to a stable dynamic system and improving the convergence. We embed a new kind of search direction by large-scale unconstrained optimization problems to overcome the efficiency of the line search. The process begins with a new expression of spectral parameter for search direction that satisfied weak-secant equations. The effectiveness relies on choosing the search direction to design algorithm with satisfied line sufficient descent conditions. The numerical experiment shows that the performance of our method is more effective in contrast to the standard initial approximation in terms of number of iteration, number of function/gradient calls and CPU times.

1. INTRODUCTION

Consider the following large scale unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x),$$

where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ twice continuously differentials and n is large. In this paper, we are interested on unconstrained optimization problem. This is about to finding the minimum of an objective function that depends on real variables.

¹corresponding author

2020 *Mathematics Subject Classification.* 90C30, 90C35.

Key words and phrases. BFGS, L-BFGS, Line search, Spectral parameter, Unconstrained optimization.

The main purpose of this research is to solve the large-scale unconstrained optimization by apply a new spectral of gradient vector via limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method. In this large-scale unconstrained optimization problem, we are interest only with a large number of n and we solve it using L-BFGS method because of its simplicity and low storage requirement. L-BFGS method also very competitive due to its low iteration cost. Some limited memory quasi-Newton formula, such as the compact L-BFGS updating formula, can preserve positive definite property based on some mild conditions [1]. Due to these remarkable advantages of the limited memory quasi-Newton formulas, they are widely utilized for large-scale optimization problems. We view the L-BFGS method as an adaptation of the BFGS method to large problem, and the implementation is almost identical to the standard BFGS method, the difference is the inverse Hessian approximation is not formed explicitly. The inverse of Hessian matrix of f is updated in this method. The L-BFGS method generates a sequence x_k by the following equation:

$$x_{k+1} = x_k + a_k d_k,$$

where $d_k = -\nabla f(x_k) = -H_k g_k$ denotes the gradient vector at x_k and a_k is the steplength. The steplength must always have a positive value such that $f(x)$ is sufficiently reduced. The success of a line search depends on the effective choices of both the search direction d_k and the step size a_k . There are a lot of formulas in calculating the step size, which are divided into an exact line search and an inexact line search. Meanwhile, H_k approximates the inverse Hessian matrix at x_k . At each iteration, x_k is a stationary point in order to get minimum or maximum value, and any new point depends on d_k and a_k . We can determine d_k first and then compute a_k through search direction and line search procedure. In the L-BFGS method, instead of forming the matrices H_k , we save the vectors s_k and y_k that define them implicitly. H_0 is given while H_k is updated according to quasi-Newton condition.

$$H_k y_k = s_k,$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$ satisfied.

Our idea is to propose our new spectral parameter of the gradient that we named as L-BFGS-QM method stand for Qadar, Muhammad Izwan and Mahani to satisfied weak secant equation. We modified the line search, αd_k , by introduced θ_k as a new spectral and we also denoted $\theta_k I$ as an approximation of Hessian H at x_k . For this

purpose, we choose to let θ_k to satisfy the weak quasi-Newton equation of [2]:

$$y_k^T H_k y_k = s_k^T y_k,$$

$$y_k^T (\theta I) y_k = s_k^T y_k = \rho_k,$$

where ρ_k is introduced and H is $n \times n$ matrix approximating the Hessian matrix which act as second derivatives and y_k^T is $n \times 1$ matrix. We will now derive a closed-form expression for θ_k . We let, $y_k = G_{k+1} s_k$, then from the left hand-side weak secant equation, we obtained,

$$s_k^T y_k = s_k^T (G_k s_k) = s_k^T (g_{k+1} - g_k) = \rho_k$$

Let $\rho_k = L(g_{k+1}, g_k, f(x_k))$, where L is Langrangian, we introduced θ_k as

$$\theta_k = \frac{\rho_k}{y_k^T y_k} = \frac{L(g_{k+1}, g_k, f(x_k))}{y_k^T y_k}.$$

2. METHODOLOGY

In this section, we propose a modification a new spectral parameter of the line search via L-BFGS method which is θ_k , that must satisfied weak secant equation in solving unconstrained optimization. Therefore, the proposed method must possess an ability to reduce the gradient components to encourage this decreasing together property, the component of $H_k = D_{k+1}$ is required. Precisely, we want to restrict the components of D_k^{-1} under some measure. For this purpose we consider, for any positive definite matrix D , the function

$$(2.1) \quad \Phi(D) = \text{tr}(D) - \ln(\det(D)),$$

where $\ln$ denotes the natural logarithm and tr is trace of matrix. Note that for all eigenvalues of D , $d_1 \geq d_2 \geq \dots \geq d_n > 0$, we have

$$(2.2) \quad \Phi(D) = \sum_{i=1}^n (d_i - \ln d_i) > 0.$$

Then, we derive an updating scheme for D_k where the components are formulated such that $\Phi(D_k)$ is minimized while satisfying to the weak quasi-Newton relation,

$$(2.3) \quad \min \text{tr}(D_{k+1}) - \ln(\det(D_{k+1}))$$

$$(2.4) \quad \text{s.t.} \quad s_k^T D_{k+1} s_k = s_k^T y_k$$

where $D_{k+1} = \text{diag} \left(D_{k+1}^{(1)}, \dots, D_{k+1}^{(n)} \right)$ and $s_k = \left(s_k^{(1)}, \dots, s_k^{(n)} \right)$. We namely (2.2) as $F(\lambda)$. Then, the minimization problem (2.1) and (2.2) becomes

$$(2.5) \quad \min \left(\sum_{i=1}^n D_{k+1}^{(i)} \right) - \ln \left(\prod_{i=1}^n D_{k+1}^{(i)} \right),$$

$$(2.6) \quad \text{s. t} \left(\sum_{i=1}^n \left(s_k^{(i)} \right)^2 D_{k+1}^{(i)} \right) - s_k^T y_k = 0.$$

Thus, the Langrangian corresponding to (2.3) and (2.4) is given by

$$(2.7) \quad L(D, \lambda) = \left(\sum_{i=1}^n D_{k+1}^{(i)} \right) - \ln \left(\prod_{i=1}^n D_{k+1}^{(i)} \right) + \lambda \left[\left(\sum_{i=1}^n \left(s_k^{(i)} \right)^2 D_{k+1}^{(i)} \right) - s_k^T y_k \right],$$

where λ is the Lagrange multiplier associated with the Lagrange function (2.5). Differentiating (2.5) partially with respect to each $D_{k+1}^{(i)}$ and setting the partial derivations to zero gives:

$$(2.8) \quad \frac{\partial L}{\partial D_{k+1}^{(i)}} = 1 - \frac{1}{D_{k+1}^{(i)}} + \lambda \left(s_k^{(i)} \right)^2 = 0, \quad i = 1, 2, \dots, n,$$

which yields

$$(2.9) \quad D_{k+1}^{(i)} = \frac{1}{1 + \lambda \left(s_k^{(i)} \right)^2}, \quad i = 1, 2, \dots, n.$$

Now, by substituting (2.6) into the constraint (2.2), we have

$$(2.10) \quad F(\lambda) = \sum_{i=1}^n \left(\frac{\left(s_k^{(i)} \right)^2}{1 + \lambda \left(s_k^{(i)} \right)^2} \right) - s_k^T y_k,$$

where the Lagrange multiplier λ can be gained by solving the nonlinear equation $F(\lambda) = 0$. Note that F is monotonically decreasing in $[0, \infty)$ since $F'(\lambda) < 0$ for all $\lambda \in [0, \infty)$, and reaches its maximum when $\lambda = 0$. Thus, (2.7) has a unique positive solution if $s_k^T s_k > s_k^T y_k$. A very high precision is neither necessary nor recommendable in the resolution of this nonlinear equation as it may lead to undesirable computation times in the resolution of the nonlinear optimization problem.

Therefore, when $s_k^T s_k > s_k^T y_k$ the Lagrange multiplier, λ_k can be approximated by

$$(2.11) \quad \begin{aligned} \lambda_k &\approx \bar{\lambda} - \frac{F(\bar{\lambda})}{F'(\bar{\lambda})} \\ &= \frac{s_k^T s_k - s_k^T y_k}{\sum_{i=1}^n \left(s_k^{(i)}\right)^4}. \end{aligned}$$

Since $\lambda > 0$, we have $D_{k+1}^{(i)} < 1$ for all $i = 1, \dots, n$ whenever $\frac{s_k^T y_k}{s_k^T s_k} < 1$. Conversely when $s_k^T s_k < s_k^T y_k$, a parameter in the form of $\frac{s_k^T y_k}{s_k^T s_k}$ would be effective in enlarging the components of D_{k+1} since $\frac{s_k^T y_k}{s_k^T s_k} > 1$. Combining the two occasions, the updating formula for D_{k+1} is given by the following:

$$(2.12) \quad D_{k+1} = \begin{cases} \text{diag} \left(D_{k+1}^{(1)}, \dots, D_{k+1}^{(n)} \right) & s_k^T s_k > s_k^T y_k \\ \frac{s_k^T y_k}{s_k^T s_k} I & \text{otherwise} \end{cases},$$

where $D_{k+1}^{(i)}$ is defined by (2.6) with λ given by (2.8).

3. CONVERGENCE ANALYSIS

In numerical analysis, a sequence of an iterative method that generated by the BFGS algorithm converge for an arbitrary initial approximation is called globally convergence. Generally, L-BFGS method is globally convergence on uniformly convex problem and its rate of convergence is $\mathbb{R}$ -linear when the identity matrix is used as initial to restart the updating scheme based on [3]. The convergence analysis that we used based on [4]. In order to analyze the global convergence property we also set up an assumptions concerning on objective function where the matrix of second derivatives of f will denoted by G which is the Hessian matrix of f . Our analysis will proceed under the following assumptions:

Assumption 1

- The objective function f is twice continuously differentiable.
- The level set $B = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is convex.
- There exist positive constants M_1 and M_2 such that

$$(3.1) \quad M_1 \|z\|^2 < z^T G(x^*) z < M_2 \|z\|^2,$$

for $\forall z \in \mathbf{R}^n$ and $\forall z \in B$ This implies that the objective function f has a unique minimize x^* in B .

The next lemma states the boundedness of $\|D_k\|$ under the assumption that D_o is taken to be the identity matrix.

Lemma 3.1. *Let x_0 be a starting point such that f satisfies Assumption 1, and $D_o = I$, where I is the $n \times n$ identity matrix. Then, for D_{k+1} defined by (2.9), the sequence $\|D_k\|$ is bounded by some positive constants c_1 and c_2 .*

Proof. See Liu and Nocedal [5]. □

Lemma 3.2. *Suppose that the assumptions in Lemma 3.1 hold. Then there exist positive constants c_3 and c_4 such that*

$$(3.2) \quad d_k^T g_k \leq -c_3 \|g_k\|^2 \text{ and } \|d_k\|^2 \leq c_4 \|g_k\|^4,$$

where $d_k = -D_k^{-1} g_k$ where D_k defined by (2.9).

A prominent feature of the updating formula (2.9) is that it generates search directions that satisfy (3.1), independent of any line search techniques. The first theorem is partly due to [6] and the second one is essentially established by [7] and can be found by [8].

Proof. Lemma 3.2 is direct result of Lemma 3.1. Thus, see Liu and Nocedal [5]. □

Theorem 3.1. *Under Assumption 1 there exist positive constants c_1 and c_2 such that, for any x_k and any d_k with $g_k^T d_k < 0$, the steplength λ_k produced by Backtracking line search (BTA) Algorithm will satisfy either*

$$(3.3) \quad f(x_k + \lambda_k d_k) - f(x_k) \leq -c_1 \frac{(g_k^T d_k)^2}{\|d_k\|^2},$$

or

$$(3.4) \quad f(x_k + \lambda_k d_k) - f(x_k) \leq c_2 g_k^T d_k.$$

Furthermore, if d_k satisfies the following conditions:

$$(3.5) \quad g_k^T d_k \leq -c_3 \|g_k\|^2 \text{ and } \|d_k\| \leq c_4 \|g_k\|$$

for some positive constants c_3 and c_4 , then

$$(3.6) \quad \liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

Proof. The first part of Theorem 3.1 is equal to Lemma 4.1 in [6]. In addition, if (3.4) hold for any d_k , then (3.2) and (3.3) become

$$(3.7) \quad f(x_k + \lambda_k d_k) - f(x_k) \leq -\frac{c_1 c_3^2}{c_4^2} \|g_k\|^2$$

and

$$(3.8) \quad f(x_k + \lambda_k d_k) - f(x_k) \leq -c_2 c_3 \|g_k\|^2,$$

since f is bounded below, we have $\liminf_{k \rightarrow \infty} f(x_k + \lambda_k d_k) - f(x_k) = 0$, which also implies (3.5). $\square$

Theorem 3.2. *Under Assumption 1, let x_k be a sequence generated by Nonmonotone Line Search (NLMS) algorithm where the considered search direction d_k satisfied the following conditions:*

$$(3.9) \quad g_k^T d_k \leq -c_1 \|g_k\|^2 \text{ and } \|d_k\| \leq c_2 \|g_k\|,$$

where c_1 and c_2 are some positive constants. Then

- the sequence x_k remains in D and every limit point x^* satisfies $g(x^*) = 0$;
- no limit point of x_k is a local maximum of f ;
- if the number of stationary points of f in D is finite, the sequence x_k converges.

4. RESULTS AND DISCUSSION

The performance of our new spectral parameter gradient vector that we proposed in weak secant equation for solving unconstrained optimization problem has been examined. We are using selected 20 test problems in Table 1 for our proposed new parameter to analyse the performance of the method. The L-BFGS routine utilized as a part of this investigation was produced by [9] with the execution of the standard L-BFGS strategies for [5]. By using Wolfe condition, the step length a_k can be find as

$$f(x_k + a_k d_k) \leq f(x_k) + c_1 a_k g(x_k)^T d_k$$

$$g(x_k)^T d_k \geq c_2 g(x_k)^T d_k,$$

with the choice $c_1 = 10^{-4}$ and $c_2 = 0.9$. The stopping condition has been set up as the number of function and gradient evaluation exceed 2000. Terminations of all the runs are active when $\|g(x_k)\| \leq 10^{-4} \times \max(1, \|(x_k)\|)$.

The performance profile [10] for a solver is the cumulative distribution function for a matrix. It is an instrument for bench-marking optimization software to access and look at the execution of optimization technique. For example, there are n_s solvers running on n_p problem, p and solver s , we determined as $f_{s,p}$ as the number of function/gradient evaluations needed to solve the problem. The number of problems in the model test set is denote as p and n_s . The performance are define by [9] as,

$$r_{s,p} = \frac{f_{s,p}}{\min(f_{s,p} : 1 \leq s \leq n_s)}.$$

The performance profile function is defined as,

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}(p : 1 \leq s \leq n_p, \log(r_{s,p} \leq \tau))$$

where $\rho_s(\tau)$ is the probability for solver s that the performance ratio $r_{s,p}$ is within a factor τ of the best possible factor. The probability that the solver will win over all the others is when $\tau = 1$. In order to get a better comparison between the results, user has to consider the value of $\rho_s(1)$. In the test, the result create by running out adjusted algorithm as the solver on the test problems and record the information in terms of number of iterations, number of function/gradient evaluation and the CPU time in second.

In this section, we will discuss about the performance of our new parameter gradient vector in LBFGS-QM method with LBFGS-I and LBFGS-MI. We have chosen two indicator in order to see the performance of our comparison which is:

1. $n = 5000, m = 3, 5$
2. $n = 10000, m = 3, 5$

where n is the number variable while m is the number of storage that we obtained. From the above indicator, we compared the performance profile based on number of iterations, number of function/gradient and CPU time in seconds.

Figures 1-6 show the performance profile in different terms of various area of interest consist of number of iterations, number of function/gradient and CPU time in seconds.

Firstly, we will focus on the number of iteration graphs. When we tested the problem with $m = 3$ and $n = 5000$ in Figure 1, it is clearly showed that our new

TABLE 1. Selected 20 test problems for comparing our proposed new parameter, LBFGS-QM with LBFGS-Identity (LBFGS-I) and LBFGS-Multiple Identity (LBFGS-MI).

Raydan 2
Extended Tridiagonal 1
Diagonal 5
Extended Maratos
Extended Cliff
Quadratic Diagonal Perturbed
Extended Wood
Exrended Quadratic Penalty QP2
Extended EP1
TRIDIA (CUTE)
ARWHEAD (CUTE)
NONDLA (CUTE)
NONQUAR (CUTE)
DQDR TIC (CUTE)
Partial Pertubed Quadratic PPQ1
Tridiagonal Pertubed Quadratic
LIARWHD (CUTE)
Diagonal 6
SINQUAD (CUTE)
Scaled Quadratic SQ2

parameter LBFGS-QM have a better performance in term of number of iteration where it take less number of iterations calls as compared to the LBFGS-MI and LBFGS-I. When the number of storage, m change to 5 it showed that LBFGS-QM have a good performance compared to LBFGS-MI and LBFGS-I. However, when the number of n increase up to 10000 at Figure 2, our new parameter is obviously showed the best performance among the existing LBFGS-MI and LBFGS-I at $m = 3$ and $m = 5$.

Next, we continued our analysis in term of number of function/gradient calls. The number of function evaluation is equal to the number of gradient evaluation because it can measure as the identical quantity as gradient is called whenever

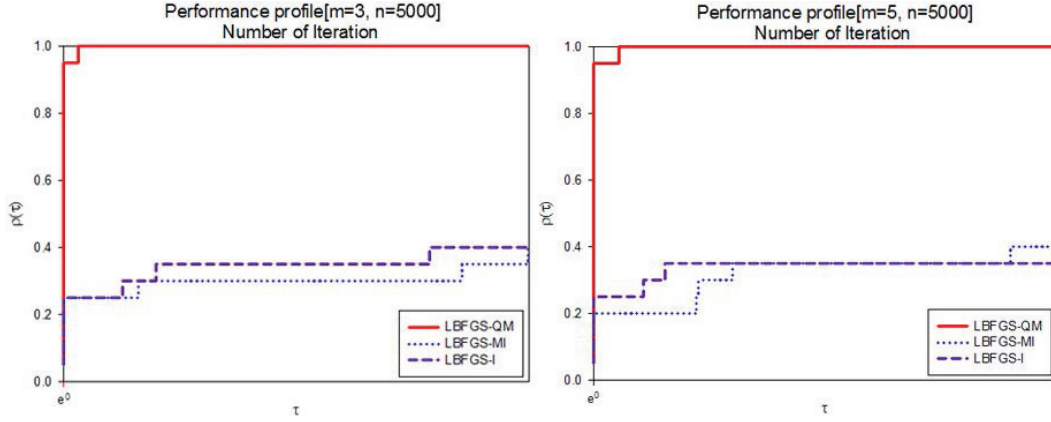


FIGURE 1. Performance profile of LBFGS-QM, LBFGS-I, and LBFGS-MI based on number of iteration for $n = 5000$.

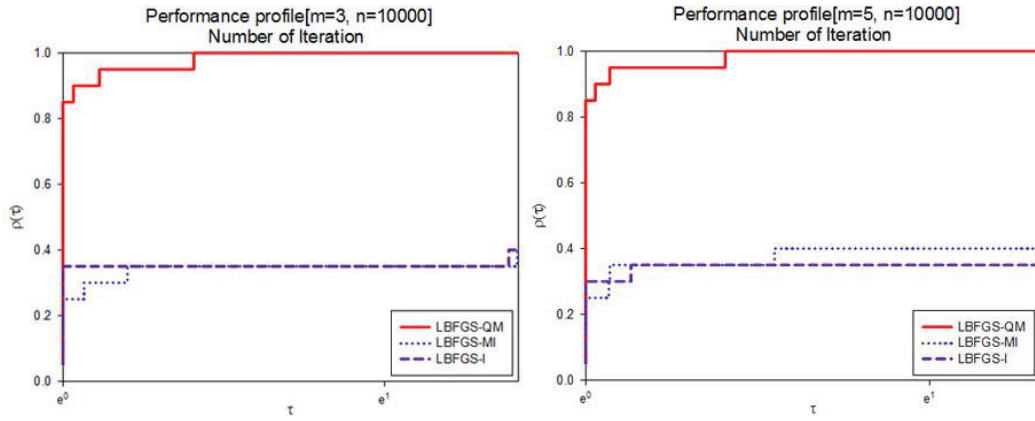


FIGURE 2. Performance profile of LBFGS-QM, LBFGS-I, and LBFGS-MI based on number of iteration for $n = 10000$.

function is called. In this case, Figure 3 show when $m = 3$ and $n = 5000$ is being used it present that LBFGS-QM is more steepest and gives good performance such require less number of function/gradient calls as compared to LBFGS-MI and LBFGS-I. When $m = 3$ and $n = 10000$, the performance is poorly at the beginning but it begin to improve after certain test problem and it is still the better performance compare to the LBFGS-MI and LBFGS-I. Next, when the m change to 5, the performance of LBFGS-QM only needs less number of function/gradient calls as compare to the LBFGS-MI and LBFGS-I while for $n = 10000$, the performance a little bit slower at the beginning, but enhanced it performance after the certain test problems.

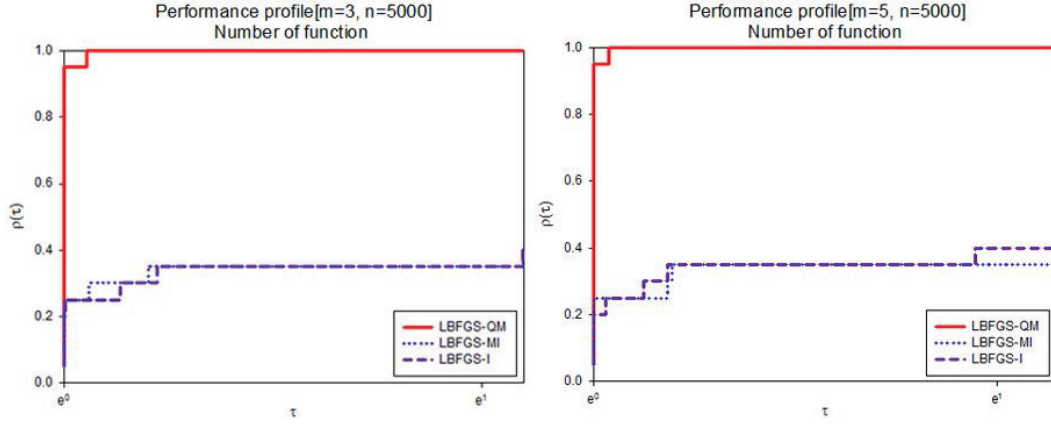


FIGURE 3. Performance profile of LBFGS-QM, LBFGS-I, and LBFGS-MI based on number of function/gradient for $n = 5000$.

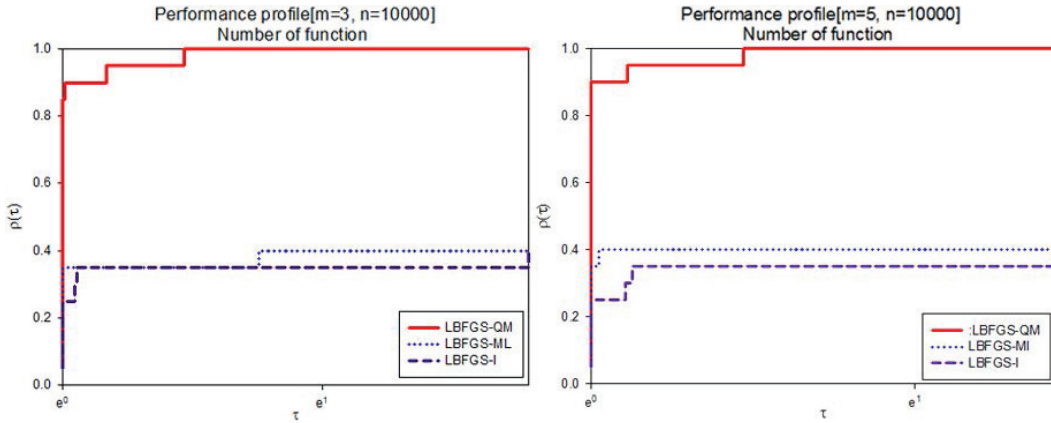


FIGURE 4. Performance profile of LBFGS-QM, LBFGS-I, and LBFGS-MI based on number of function/gradient for $n = 10000$.

Apart from the number of iteration and the number of function/gradient evaluation, we also focus on CPU time in seconds. The CPU time is analyzed based on second and its duration includes times needed to generate search direction with specific end goal to perform line search and convergence test. Figure 5 showed that the performance of our LBFGS-QM take a shorter time to converge compared with LBFGS-MI and LBFGS-I. In addition, same goes with Figure 6 when we change n from 5000 to 10000, LBFGS-QM only required shorter time as compare to LBFGS-MI and LBFGS-I.

The experiment shows that our proposed parameter have an excellent performance over existing LBFGS-MI and LBFGS-I. The figures evidently display the performance profile of our proposed parameter that are in outstanding state compared to the existing LBFGS-MI and LBFGS-I. These demonstrate that our new parameter is the best performer compared to the LBFGS-MI and LBFGS-I.

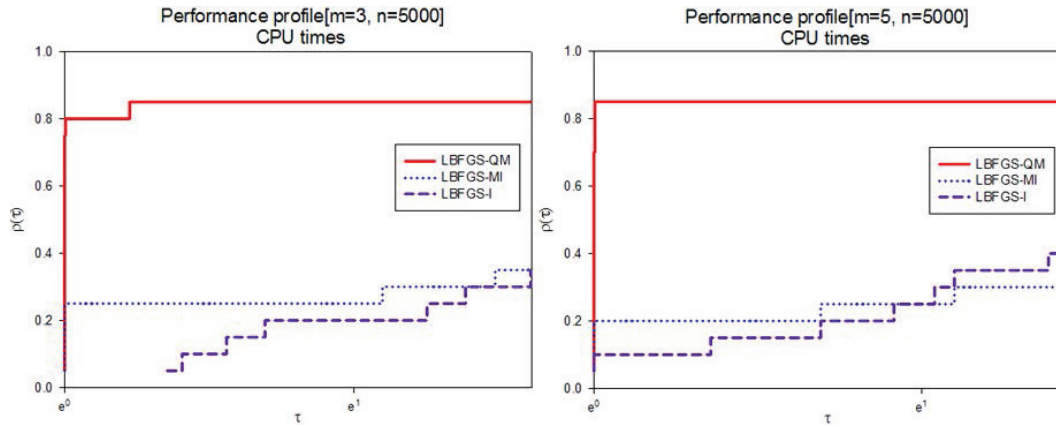


FIGURE 5. Performance profile of LBFGS-QM, LBFGS-I, and LBFGS-MI based on CPU times for $n = 5000$.

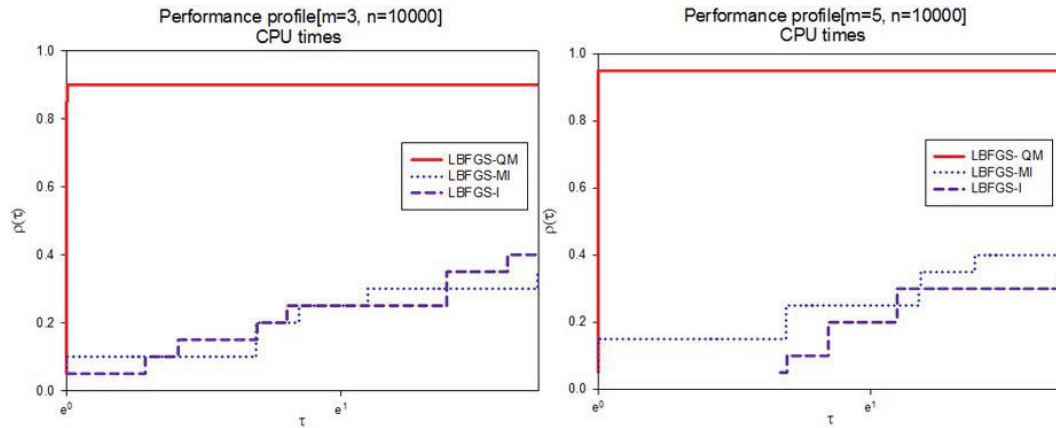


FIGURE 6. Performance profile of LBFGS-QM, LBFGS-I, and LBFGS-MI based on CPU times for $n = 10000$.

5. CONCLUSION

In this paper, we interested to evaluate our new spectral parameter which is LBFGS-QM. These variant of spectral parameter are able to display it efficiency and effectiveness in reducing the number of iterations, number of function/gradient and CPU time in seconds. It will be interesting if we diversify the scope to use larger number of storage such as $n = 50000$ and compared the result with the current research and find the way to increase the effectiveness of the research.

ACKNOWLEDGEMENT

The researchers of this paper would appreciate the supports of Universiti Putra Malaysia through Putra Grant No. Vot 9544900.

REFERENCES

- [1] J.E. DENNIS, J.J. MORÉ,: *Quasi-Newton methods, motivation and theory*, SIAM review, **19**(1), 46-89, 1977.
- [2] J.E. DENNIS, H. WOLKOWICZ: *Sizing and least-change secant methods*, SIAM Journal on Numerical Analysis, **30**(5), 1291-1314, 1993.
- [3] J. NOCEDAL, S.J. WRIGHT: *Large-scale unconstrained optimization*, Numerical Optimization, 164-192, 1996.
- [4] H.S. SIM, W.J. LEONG, AND C.Y. CHEN: *Gradient method with multiple damping for large-scale unconstrained optimization*, Optimization Letters, **13** (2019), 617–632.
- [5] D.C. LIU, J. NOCEDAL: *On the limited memory bfgs method for large scale optimization*, Mathematical programming, **45**(1-3) (1989), 503-528.
- [6] R.H. BYRD, J. NOCEDAL: *A tool for the analysis of quasi-newton methods with application to unconstrained minimization*, SIAM Journal on Numerical Analysis, **26**(3) (1989), 727-739.
- [7] L. GRIPPO, F. LAMPARIELLO, S. LUCIDI: *A nonmonotone line search technique for newtons method*, SIAM Journal on Numerical Analysis, **23**(4) (1986), 707-716.
- [8] F. LUENGO, M. RAYDAN, W. GLUNT, T.L. HAYDEN: *Preconditioned spectral gradient method*, Numerical Algorithms, **30**(3-4) (2002), 241-258.
- [9] J. NOCEDAL: *Updating quasi-newton matrices with limited storage*, Mathematics of computation, **35**(151) (1980), 773-782.
- [10] E.D. DOLAN, J.J. MORÉ,: *Benchmarking optimization software with performance profiles*, Mathematical programming, **91**(2) (2002), 201-213.

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE,
UNIVERSITI PUTRA MALAYSIA, 43400 SERDANG, SELAGOR, MALAYSIA.
Email address: muhdizwanishak@gmail.com

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE,
UNIVERSITI PUTRA MALAYSIA, 43400 SERDANG, SELAGOR, MALAYSIA.
Email address: smahani@upm.edu.my

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE,
UNIVERSITI PUTRA MALAYSIA, 43400 SERDANG, SELAGOR, MALAYSIA.
Email address: leongwj@upm.edu.my